Fast Reconstruction of Compact Context-Specific Metabolic Network Models

Nikos Vlassis* Maria Pires Pacheco† Thomas Sauter†

*Luxembourg Centre for Systems Biomedicine and †Life Sciences Research Unit
University of Luxembourg

May 1, 2013

Abstract

Systemic approaches to the study of a biological cell or tissue rely increasingly on the use of context-specific metabolic network models. The reconstruction of such a model from high-throughput data can routinely involve large numbers of tests under different conditions and extensive parameter tuning, which calls for fast algorithms. We present FASTCORE, a generic algorithm for reconstructing context-specific metabolic network models from global genome-wide metabolic network models such as Recon X. FASTCORE takes as input a core set of reactions that are known to be active in the context of interest (e.g., cell or tissue), and it searches for a flux consistent subnetwork of the global network that contains all reactions from the core set and a minimal set of additional reactions. Our key observation is that a minimal consistent reconstruction can be defined via a set of sparse modes of the global network, and FASTCORE iteratively computes such a set via a series of linear programs. Experiments on liver data demonstrate speedups of several orders of magnitude, and significantly more compact reconstructions, over a chief rival method. Given its simplicity and its excellent performance, FASTCORE can form the backbone of many future metabolic network reconstruction algorithms.

1 Introduction

Cell metabolism is known to play a key role in the pathogenesis of various diseases [11] such as Parkinson's disease [28] and cancer [18]. The study of human metabolism has been greatly advanced by the development of computational models of metabolism, such as Recon 1 [12], the Edinburgh human metabolic network [17], and Recon 2 [37]. These are genome-scale metabolic network models that have been reconstructed by combining various sources of 'omics' and literature data, and they involve a large set of biochemical reactions that can be active in different contexts, e.g., different cell types or tissues [36].

To maximize the predictive power of a metabolic model when conditioning on a specific context, for instance the energy metabolism of a neuron or the metabolism of liver, recent efforts go into the development of *context-specific* metabolic models [9, 21, 8, 23, 2]. These are network models that are derived from global models like Recon 1, but they only contain

a subset of reactions, namely, those reactions that are active in the given context. Such context-specific metabolic models are known to exhibit superior explanatory and predictive power than their global counterparts [21, 14, 5].

Most algorithms for context-specific metabolic network reconstruction first identify a relevant subset of reactions according to some 'omics' information (typically expression data and bibliomics), and then search for a subnetwork of the global network that satisfies some mathematical requirements and contains all (or most of) these reactions [3, 32, 21, 7, 19, 2]. The mathematical requirements are typically imposed via flux balance analysis, which characterizes the steady-state distribution of fluxes in a metabolic network via linear constraints that are derived from the stoichiometry of the network and physical conservation laws [31, 34, 29, 15, 13]. The search problem may target the optimization of a specific functionality of the model (e.g., biomass production) or some other objective [4], and it may involve repeated tests under different conditions and parameter tuning [3, 14, 27]. The latter calls for fast algorithms.

We present FASTCORE, a generic algorithm for context-specific metabolic network reconstruction. FASTCORE takes as input a core set of reactions that are supported by strong evidence to be active in the context of interest. Then it searches for a flux consistent subnetwork of the global network that contains all reactions from the core set and a minimal set of additional reactions. Flux consistency implies that each reaction of the network is active (i.e., has nonzero flux) in at least one feasible flux distribution [31, 1]. An attractive feature of FASTCORE is its generality: As it only relies on a preselected set of reactions and a simple mathematical objective (flux consistency), it can be applied in different contexts and it allows the integration of different pieces of evidence ('multi-omics') into a single model.

Computing a minimal consistent reconstruction from a subset of reactions of a global network is, however, an NP-hard problem [1], and hence some approximation is in order. Our key observation is that a minimal consistent reconstruction can be defined via a set of *sparse* modes of the global network, and FASTCORE is designed to compute a minimal such set. Every iteration of the algorithm computes a new sparse mode via two linear programs that aim at maximizing the support of the mode inside the core set while minimizing that quantity outside the core set. FASTCORE's search strategy is in marked contrast to related approaches, in which the search for a minimal consistent reconstruction involves, for instance, incremental network pruning [21]. FASTCORE is simple, devoid of free parameters, and its performance is excellent in practice: As we demonstrate on experiments with liver data, FASTCORE is several orders of magnitude faster, and produces much more compact reconstructions, than the main competing algorithm MBA [21].

2 Methods

2.1 Background

A metabolic network of m metabolites and n reactions is represented by an $m \times n$ *stoichiometric* matrix S, where each entry S_{ij} contains the stoichiometric coefficient of metabolite i in reaction j. A *flux* vector $v \in \mathbb{R}^n$ is a tuple of reaction rates, $v = (v_1, \ldots, v_n)$, where v_i

is the rate of reaction i in the network. Reactions are grouped into *reversible* ones ($\mathcal{R}$) and *irreversible* ones ($\mathcal{I}$). For a reaction $i \in \mathcal{I}$ holds $v_i \geq 0$; this and other imposed flux bounds, e.g., lower and upper bounds per reaction, are denoted by $\mathcal{B}$ (which defines a convex set). A flux vector is called *feasible* or a *mode* if it satisfies a set of steady state mass-balance constraints that can be compactly expressed as:

$$Sv = 0, \quad v \in \mathcal{B}\,. \tag{1}$$

An *elementary* mode is a feasible flux vector $v \neq 0$ with minimal support; that is, there is no other feasible flux vector $w \neq 0$ with $supp(w) \subset supp(v)$, where $supp(v)$ denotes the support (i.e., the set of nonzero entries) of v [31, 15]. A metabolic network model is called *(flux) consistent* if each reaction in the network is in the support of some mode, that is, for each reaction i there exists a mode v with $v_i \neq 0$ (in practice $|v_i| \geq \varepsilon$, for some small positive threshold ε) [31, 1].

2.2 Network consistency testing

Given a metabolic network model with stoichiometric matrix S, a problem of interest is to test whether the network is consistent or not. Additionally, if the network is inconsistent, it would be desirable to have a method that detects the inconsistent part.

It has been suggested that network consistency can be detected by a single linear program (LP) [1]. The idea is to first convert each reversible reaction into two irreversible reactions (and define a reversible flux as the difference of two irreversible fluxes), and then test if the minimum feasible flux on the new set $\mathcal{J}$ of irreversible-only reactions is strictly positive (in practice, at least ε). This is equivalent to testing if the following LP is feasible:

$$\begin{aligned} \max_{v,z} \quad & z & & \\ \text{s.t.} \quad & z \geq \varepsilon & & z \in \mathbb{R} \\ & v_i \geq z & & \forall i \in \mathcal{J} \\ & Sv = 0 & & v \in \mathcal{B}\,. \end{aligned} \tag{LP-2}$$

This test of consistency, however, can produce spurious solutions. In Figure 1 we show a toy metabolic network comprising four metabolites (A,B,C,D) and six reactions annotated with corresponding fluxes $v_1, \ldots, v_6$, each bounded by $|v_i| \leq 3$. All stoichiometric coefficients are equal to one, except for the reaction $\rightarrow$2A. The only reversible reaction is A$\leftrightarrow$B, which is a dead-end reaction and therefore inconsistent, whereas all other reactions are irreversible and consistent. After converting A$\leftrightarrow$B to a pair of irreversible reactions, LP-2 achieves optimal value $z^* = 1.5$, which implies (wrongly) that the network is consistent. The test here fails because the two irreversible copies of A$\leftrightarrow$B have equal flux at the solution, thereby nullifying the actual net flux of A$\leftrightarrow$B.

A straightforward solution to the problem would involve iterating through all reactions, computing the maximum and minimum feasible flux of each reaction via an LP that satisfies the constraints in (1). This is the idea behind the FVA algorithm and the *reduceModel* function of the COBRA toolbox [24, 30]. However, iterating through all reactions can be

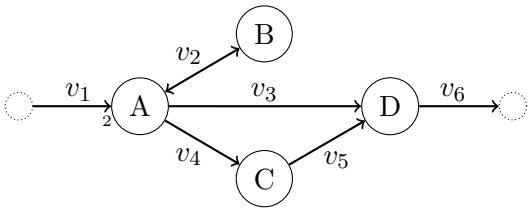

Figure 1: A metabolic network with one inconsistent reaction (A↔B)

inefficient. A faster variant is fastFVA [16], which achieves acceleration over FVA via LP warm-starts. Another fast algorithm is CMC [21], which involves a series of LPs, where each LP maximizes the sum of fluxes over a subset $\mathcal{J}$ of reactions:

$$\begin{aligned} \max_{v} \quad & \sum_{j \in \mathcal{J}} v_j \\ \text{s.t.} \quad & Sv = 0 \qquad v \in \mathcal{B}\,. \end{aligned} \tag{LP-3}$$

The set $\mathcal{J}$ is initialized by $\mathcal{J} = \mathcal{R} \cup \mathcal{I}$ (all reactions in the network), and it is updated after each run of LP-3 so that it contains the reactions whose consistency has not been established yet. When $\mathcal{J}$ cannot be reduced any further, we can reverse the signs of the columns of S corresponding to the reversible reactions in $\mathcal{J}$ and resume the iterations. Eventually, all remaining reactions may have to be tested one by one for consistency, as in FVA. Such an iterative scheme is complete, in the sense that it will always report consistency if the network is consistent, and if not, it will reveal the set of inconsistent reactions. However, as we will clarify in the next section, LP-3 is not optimizing the 'correct' function, which may result in unnecessarily many iterations. For example, when applied to the network of Figure 1, LP-3 will pick up the elementary mode that corresponds to the pathway A→C→D (because this pathway achieves maximum sum of fluxes $v_1 + v_4 + v_5 + v_6 = 1.5 + 3 + 3 + 3$), and it will set $v_3 = 0$. To establish the consistency of the reaction A→D, an additional run of LP-3 would be needed, where the set $\mathcal{J}$ would only involve the reactions A↔B and A→D. Hence, an iterative algorithm like CMC that relies on LP-3 would need two iterations to detect the consistent part of this network. However, one LP suffices in this example, as we explain in the next section. In more general problems involving larger and more realistic networks, CMC may involve unnecessarily many iterations, as we demonstrate in the experiments.

2.3 Fast detection of consistent reactions

In most problems of interest there will be no single mode that renders the whole network consistent, and an iterative algorithm like the one described in the previous section must be used. For performance reasons it would therefore be desirable to be able to establish the consistency of as many reactions as possible in each iteration of the algorithm.

Since consistency implies nonzero fluxes, it is sufficient to optimize a function that just 'pushes' all fluxes away from zero. Formally, this amounts to searching for modes v whose *cardinality*—denoted by $card(v)$ and defined as $card(v) = |supp(v)|$, i.e., the number of nonzero entries of v—is as large as possible. Directly maximizing $card(v)$ is, however, not

straightforward. First, the *card* function is quasiconcave only for $v \in \mathbb{R}^n_+$ (the nonnegative orthant), and it is nonconvex for general $v \in \mathbb{R}^n$ [6]. Second, even if we restrict attention to nonnegative fluxes in each iteration (which we can do without loss of generality by flipping the signs of the corresponding columns of S), it is not obvious how to efficiently maximize the quasiconcave $card(v)$. Third, in practice consistency implies fluxes that are ε-distant from zero, in which case some adaptation of the *card* function is in order.

The above suggests the following approximation to the *card* function for nonnegative fluxes. Note that, when $v \geq 0$, the cardinality function can be expressed as

$$card(v) = \sum_{i=1}^{n} \theta(v_i)\,, \tag{4}$$

where $\theta : \mathbb{R} \to \{0, 1\}$ is a step function:

$$\theta(v_i) = \begin{cases} 0 & \text{if } v_i = 0 \\ 1 & \text{if } v_i > 0\,. \end{cases} \tag{5}$$

The key idea is to approximate the function θ by a concave function that is the minimum of a linear function and a constant function:

$$\theta(v_i) \approx \min\{\frac{v_i}{\varepsilon}, 1\}\,, \tag{6}$$

where ε is the flux threshold. Introducing an auxiliary nonnegative variable $z_i \in \mathbb{R}_+$ for each flux variable v_i, and taking epigraphs [6], it is easy to verify that, for constant ε, the function $card(v)$ can be approximately maximized over an arbitrary set $\mathcal{J}$ of nonegative fluxes via the following LP:

$$\begin{array}{lll} \max\limits_{v,z} & \sum\limits_{i \in \mathcal{J}} z_i & \\ \text{s.t.} & z_i \in [0, \varepsilon] & \forall i \in \mathcal{J},\ z_i \in \mathbb{R}_+ \\ & v_i \geq z_i & \forall i \in \mathcal{J} \\ & Sv = 0 & v \in \mathcal{B}\,. \end{array} \tag{LP-7}$$

Note that LP-7 tries to maximize the number of feasible fluxes in $\mathcal{J}$ whose value is at least ε (contrast this with LP-2).

Returning to the network of Figure 1, if $\mathcal{J}$ comprises all network reactions, then note that the flux vector $[v_1, v_2, v_3, v_4, v_5, v_6] = [\varepsilon, 0, \varepsilon, \varepsilon, \varepsilon, 2\varepsilon]$ is an optimal solution of LP-7. Hence, a single run of the latter can establish the consistency of all reactions (except v_2) of that network. More generally, a single run of LP-7 on an arbitrary subset $\mathcal{J}$ of a given network will typically identify all consistent irreversible reactions of $\mathcal{J}$. The intuition is that LP-7 prefers flux 'splitting' over flux 'concentrating' in order to maximize the number of participating reactions in the solution, which, in the case of irreversible reactions, corresponds to flux cardinality maximization.

By construction, the above approximation of the cardinality function applies only to nonnegative fluxes. In order to deal with reversible reactions that can also take negative fluxes, we can embed LP-7 in an iterative algorithm (as in the previous section), in which reversible

reactions are first considered for positive flux via LP-7, and then they are considered for negative flux. The latter is possible by flipping the signs of the columns of the stoichiometric matrix that correspond to the reversible reactions under testing, in which case the fluxes of the transformed model are again all nonnegative, and the above approximation of the cardinality function can be used. This gives rise to an algorithm for detecting the consistent part of a network that we call FASTCC (for *fast* consistency *c*heck). Since FASTCC is just a variant of FASTCORE, we defer its detailed description until the next section.

2.4 Context-specific network reconstruction

The reconstruction problem involves computing a minimal consistent network from a global network $\mathcal{N}$ (with stoichiometric matrix S) and a 'core' set $\mathcal{C}$ of reactions that are known to be active in a given context. Formally, given a consistent network $\mathcal{N}$ and a set $\mathcal{C} \subset \mathcal{N}$, the problem is to find the smallest consistent subnetwork $\mathcal{A} \subseteq \mathcal{N}$ such that $\mathcal{C} \subseteq \mathcal{A}$. This problem is known to be NP-complete[1] [1], suggesting that a practical solution should entail some approximation.

Our approach hinges on the observation that a consistent subnetwork $\mathcal{A}$ of $\mathcal{N}$ can be defined via a set of modes of $\mathcal{N}$:

Theorem 1. *Let $\mathcal{V}$ be a set of modes of $\mathcal{N}$, and let $\mathcal{A} = \cup_{v \in \mathcal{V}} supp(v)$ be the union of the supports of these modes. The network model defined by $\mathcal{A}$ and $S_\mathcal{A}$, the submatrix of S that contains only the columns indexed by $\mathcal{A}$, is consistent.*

Proof. For each $v \in \mathcal{V}$, let $v_\mathcal{A}$ be the 'truncated' v after dropping all dimensions not indexed by $\mathcal{A}$. Clearly, $S_\mathcal{A} v_\mathcal{A} = 0$, therefore each $v_\mathcal{A}$ is a mode in the reduced model $\{\mathcal{A}, S_\mathcal{A}\}$. By construction of $\mathcal{A}$, each reaction in $\mathcal{A}$ is in the support of some $v \in \mathcal{V}$, and hence also in the support of some mode $v_\mathcal{A}$ of the reduced model. □

This simple result allows one to cast the reconstruction problem as a search problem over sets of modes of the global network $\mathcal{N}$:

$$\begin{aligned} \min_{\mathcal{V}} \quad & card(\mathcal{A}) \\ \text{s.t.} \quad & \mathcal{A} = \bigcup_{v \in \mathcal{V}} supp(v) \\ & \mathcal{C} \subseteq \mathcal{A}\,. \end{aligned} \tag{8}$$

This is still a difficult (combinatorial optimization) problem, but it is amenable to useful approximations. In particular, in FASTCORE we search over sets of modes $\mathcal{V}$ via a greedy scheme that incrementally adds modes to $\mathcal{V}$, reminiscent of greedy heuristics for the related *set covering problem* [10]. Further, as a means to approximately minimize $card(\mathcal{A})$, each

[1] Acuña et al. [1] prove NP-completeness of this problem by noting that a special case involves $\mathcal{C}$ being the empty set, in which case the problem comes down to finding the smallest elementary mode of $\mathcal{N}$, which is NP-complete [1]. However, this leaves open the case of a nonempty core set $\mathcal{C}$, in which a solution need not constitute an elementary mode. We conjecture that the problem remains NP-hard when $\mathcal{C}$ is nonempty, but we are not pursuing this question here.

Input: A consistent metabolic network $\mathcal{N} = \mathcal{R} \cup \mathcal{I}$ with stoichiometric matrix S, and a set $\mathcal{C} \subset \mathcal{N}$.
Output: A consistent subnetwork $\mathcal{A} \subseteq \mathcal{N}$ such that $\mathcal{C} \subseteq \mathcal{A}$.

function FASTCORE($\mathcal{N}, \mathcal{C}$)
  $\mathcal{J} \leftarrow \mathcal{C} \cap \mathcal{I}$, $\mathcal{P} \leftarrow \mathcal{N} \setminus \mathcal{C}$, $\mathcal{A} \leftarrow \emptyset$
  $flipped \leftarrow false$, $singleton \leftarrow false$
  $\mathcal{A} \leftarrow \mathcal{A} \cup$ FINDSPARSEMODE($\mathcal{J}, \mathcal{P}, false$)
  $\mathcal{J} \leftarrow \mathcal{C} \setminus \mathcal{A}$
  while $\mathcal{J} \neq \emptyset$ **do**
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{A}$
    $\mathcal{A} \leftarrow \mathcal{A} \cup$ FINDSPARSEMODE($\mathcal{J}, \mathcal{P}, singleton$)
    if $\mathcal{J} \cap \mathcal{A} \neq \emptyset$ **then**
      $\mathcal{J} \leftarrow \mathcal{J} \setminus \mathcal{A}$, $flipped \leftarrow false$
    else
      if $flipped$ **then**
        $flipped \leftarrow false$, $singleton \leftarrow true$
      else
        $flipped \leftarrow true$
        if $singleton$ **then**
          $\tilde{\mathcal{J}} \leftarrow \mathcal{J}(1)$ (the first element of $\mathcal{J}$)
        else
          $\tilde{\mathcal{J}} \leftarrow \mathcal{J}$
        end if
        for each $i \in \tilde{\mathcal{J}} \setminus \mathcal{I}$ **do**
          flip the sign of the i'th column of S and swap the upper and lower bounds of v_i
        end for
      end if
    end if
  end while
  return $\mathcal{A}$
end function

Input: A set $\mathcal{J} \subseteq \mathcal{C}$, a penalty set $\mathcal{P} \subseteq \mathcal{N} \setminus \mathcal{C}$, and the *singleton* flag (s).
Output: The support of a $\mathcal{P}$-sparse mode.

function FINDSPARSEMODE($\mathcal{J}, \mathcal{P}, s$)
  if $\mathcal{J} = \emptyset$ **then**
    return $\emptyset$
  end if
  if $s = true$ **then**
    $v^* \leftarrow$ LP-7 on set $\mathcal{J}(1)$
  else
    $v^* \leftarrow$ LP-7 on set $\mathcal{J}$
  end if
  $\mathcal{K} \leftarrow \{i \in \mathcal{J} : v_i^* \geq \varepsilon\}$
  if $\mathcal{K} = \emptyset$ **then**
    return $\emptyset$
  end if
  $v^* \leftarrow$ LP-9 on sets $\mathcal{K}, \mathcal{P}$
  return $\{i \in \mathcal{N} : |v_i^*| \geq \varepsilon\}$
end function

Figure 2: The FASTCORE algorithm for context-specific metabolic network reconstruction.

added mode is constrained to have *sparse* support outside $\mathcal{C}$. This is implemented via L_1-norm minimization, which is a standard approach to computing sparse solutions to (convex) optimization problems [6, 22].

The overall FASTCORE algorithm is shown in Figure 2. The algorithm maintains a set $\mathcal{J} \subseteq \mathcal{C}$ that is initialized with the irreversible reactions in $\mathcal{C}$, and a 'penalty' set $\mathcal{P} = (\mathcal{N} \setminus \mathcal{C}) \setminus \mathcal{A}$ that contains all reactions outside $\mathcal{C}$ that have not been added yet to the set $\mathcal{A}$. Each iteration adds to the set $\mathcal{A}$ the support of a mode that is sparse in $\mathcal{P}$, computed by the function FINDSPARSEMODE. The latter first involves an LP-7 to compute an active subset $\mathcal{K}$ of $\mathcal{J}$,

and then the following L_1-norm minimization LP constrained by the set $\mathcal{K}$:

$$\begin{aligned} \min_{v,z} \quad & \sum_{i\in\mathcal{P}} z_i \\ \text{s.t.} \quad & v_i \in [-z_i, z_i] && \forall i \in \mathcal{P},\ z_i \in \mathbb{R}_+ \\ & v_i \geq \varepsilon && \forall i \in \mathcal{K} \\ & Sv = 0 && v \in \mathcal{B}\,. \end{aligned} \tag{LP-9}$$

The LP-9 minimizes $\sum_{i\in\mathcal{P}} |v_i|$, the L_1 norm of fluxes in the penalty set $\mathcal{P}$ (expressed via epigraphs), subject to a minimum flux constraint on the set $\mathcal{K}$.[2] The algorithm first goes through the $\mathcal{I} \cap \mathcal{C}$ reactions, then the $\mathcal{R} \cap \mathcal{C}$ ones, and eventually through each individual reaction (variable *singleton*). The *flipped* variable ensures that a reversible reaction is tested in both the forward and negative direction. The algorithm terminates when all reactions in $\mathcal{C}$ have been added to $\mathcal{A}$, which is guaranteed since in the main loop the set $\mathcal{J}$ never expands (step 10) and the global network $\mathcal{N}$ is consistent. Note that FASTCORE has no free parameters besides the flux threshold ε.

The FASTCC algorithm for detecting the consistent part of an input network (see previous section) can be viewed as a variant of FASTCORE$(\mathcal{N}, \mathcal{N})$ in which the steps 10–14 of FINDSPARSEMODE are omitted (and there is no $\mathcal{P}$ set). It is easy to verify that FASTCC is complete, in the sense that it will always report consistency if the network is consistent, and if not, it will reveal the set of inconsistent reactions.

The closest algorithm to FASTCORE is the MBA algorithm of Jerby et al. [21]. MBA takes as input two core sets of reactions, and it searches for a consistent network that contains all reactions from the first set, a maximum number of reactions from the second set (for a given tradeoff), and a minimal number of reactions from the global network.[3] Both FASTCORE and MBA involve a search for a minimal consistent subnetwork, however the search strategy of FASTCORE is very different to MBA: Whereas FASTCORE iteratively expands the active set $\mathcal{A}$ starting with $\mathcal{A} = \emptyset$, MBA starts with $\mathcal{A} = \mathcal{N}$ and iteratively prunes the set $\mathcal{A}$ by checking whether the removal of each individual reaction (selected in random order) compromises network consistency. Consistency testing in MBA is carried out with the CMC algorithm that is based on LP-3, as explained earlier. Hence, FASTCORE's search strategy differs to MBA in two key aspects: First, consistency testing in FASTCORE involves the maximization of flux cardinality (LP-7) instead of sum of fluxes (LP-3), which results in fewer LP iterations (see experiments). Second, the search for compact solutions in FASTCORE involves L_1-norm

[2] Some care is needed to preempt false negative solutions arising from the minimization of L_1 norm in LP-9. For example, suppose in the network of Figure 1 that $\mathcal{N}$ comprises all reactions except A↔B, and $\mathcal{C} = \mathcal{J} = \mathcal{K} = \{6\}$ and $\mathcal{P} = \{1, 3, 4, 5\}$. In this case, LP-9 could settle to a solution $[v_1, v_3, v_4, v_5, v_6] = [\frac{\varepsilon}{2}, \varepsilon, 0, 0, \varepsilon]$. The flux v_1, being below ε, would be treated as zero by FINDSPARSEMODE, in which case the reaction →2A would be erroneously excluded from the reconstruction. A simple way to avoid this is to use a scaled version of ε (we used $10^5\varepsilon$) in the second constraint of LP-9, with an equal scaling of all flux bounds in $\mathcal{N}$.

[3] FASTCORE can be easily adapted to work with multiple core sets, by introducing a set of weights that reflect the confidence of each reaction to be active in the given context, and adding appropriate regularization terms in the objective functions of LP-7 and LP-9 that capture the given tradeoff. We will address this variant in future work.

Table 1: Comparing FASTCC to fastFVA [16] and CMC [21] on three input models

	c-Mouse ($\|\mathcal{N}\| = 2432$)		c-Recon1 ($\|\mathcal{N}\| = 2469$)		c-Recon2 ($\|\mathcal{N}\| = 5834$)	
	# LPs	time[4]	# LPs	time	# LPs	time
fastFVA	4864	9	4938	9	11668	207
CMC	184	11	49	2	42	11
FASTCC	2	0.2	9	0.4	19	5

minimization instead of pruning. The advantage of the former is that it can be encoded by a single LP, resulting in significant speedups.

3 Results

We report results on two sets of problems, the first involving consistency verification of an input model, and the second involving the reconstruction of a context-specific model from an input model and a core set of reactions. The FASTCORE algorithm was implemented in the COBRA toolbox [30], using Matlab 2013a and the IBM CPLEX solver (version 12.5.0.0). Test runs were performed on a standard 1.8 GHz Intel Core i7 laptop with 4 GB RAM running Mac OS X 10.7.5. In all experiments we used flux threshold ε =1e-4. The software is available from `http://bio.uni.lu/systems_biology/software/fastcore/`

3.1 Consistency testing

In the first set of experiments we applied FASTCC, the consistency testing variant of FASTCORE, for consistency verification of three input models, and compared it against the FastFVA algorithm of Gudmundsson and Thiele [16], and an own implementation (based on FASTCC but with LP-3 replacing LP-7) of the CheckModelConsistency (CMC) algorithm of Jerby et al. [21]. We also tested the FVA algorithm of the *reduceModel* function of the COBRA toolbox [30], and the MIRAGE algorithm of Vitkin and Shlomi [38], but we do not include them in the results as they performed worse than the reported ones. The three input models were the following:

- c-Mouse ($|\mathcal{N}| = 2432$), the consistent part (after removing all inconsistent reactions, in total 1295) of the genome-wide metabolic model of mouse, developed by Sigurdsson et al. [33].
- c-Recon1 ($|\mathcal{N}| = 2469$), the consistent part of Recon 1 [12]. (Recon 1 was found to contain 1273 inconsistent reactions.)
- c-Recon2 ($|\mathcal{N}| = 5834$), the consistent part of Recon 2 [37]. (Recon 2 was found to contain 1606 inconsistent reactions.)

[4] in seconds

Table 2: Comparing FASTCORE to MBA [21] on liver model reconstruction from c-Recon1

	liver core set ($\|\mathcal{C}\| = 1069$)				strict liver core set ($\|\mathcal{C}\| = 1083$)			
	$\|\mathcal{A}\|$	IR[5]	#LPs	time[6]	$\|\mathcal{A}\|$	IR	#LPs	time
MBA	1826	1573	72279	7383	1888	1630	71546	6730
FASTCORE	1746	1546	20	1	1818	1627	20	1

The results are shown in Table 1. FASTCC is using a significantly smaller number of LPs than the other two algorithms, and is in general faster. We note that fastFVA is based on a highly optimized Matlab/C++ implementation with LP warm-starts, while FASTCC is based on standard Matlab. These results confirm the appropriateness of flux cardinality (LP-7) as a metric for network consistency testing, in agreement with the theoretical analysis and the discussions above.

3.2 Reconstruction of a liver model

In the second set of experiments, we used the FASTCORE algorithm to reconstruct a liver specific metabolic network model from the consistent part of Recon 1 (c-Recon1, $|\mathcal{N}| = 2469$), and we compared against an own implementation of the MBA algorithm of Jerby et al. [21]. We applied the two algorithms in two settings. The first setting involves the liver specific input reaction set of Jerby et al. [21], which is based on 779 'high' core and 290 'medium' core reactions (the latter set is supported by weaker biological evidence than the former). To allow a comparison with FASTCORE, we defined a single core set as the union of the high and medium core reaction sets, and we applied the two algorithms on this core set. The second setting uses the 'strict' liver model of Jerby et al. [21], which contains 1083 high core reactions and no medium core reactions, and therefore allows a direct comparison with FASTCORE.

The results for the two settings are shown in Table 2 (note that for MBA, the reported number of LPs and the runtime refer to a single pruning iteration of the algorithm; the corresponding figures for the full MBA algorithm could be 1000-fold higher). In both settings, FASTCORE is several orders of magnitude faster than MBA, achieving a full reconstruction of a liver specific model in about one second, using a much smaller number of LPs. As MBA employs a greedy pruning strategy for optimization, the number of LPs that it uses and its total runtime can be very high, as also indicated by Wang et al. [39] who reported runtime of a single pruning pass of MBA in the order of 10 hours on a 2.34 GHz CPU computer. The reconstructed models by FASTCORE are also considerably more compact than those obtained by MBA, with a difference of 70-80 non-core reactions. For the liver model, 1683 out of the 1746 reactions (96%) of the FASTCORE reconstruction appear also in the MBA reconstruction, whereas for the strict liver model the common reactions are 1743 out of 1818 (also 96%).

To evaluate FASTCORE's performance in correctly identifying liver reactions, we performed

[5] number of intracellular reactions

[6] the reported time (in seconds), as well as the number of LPs, refer to a single pruning step of MBA; the runtime of the complete MBA algorithm would be 1000-fold.

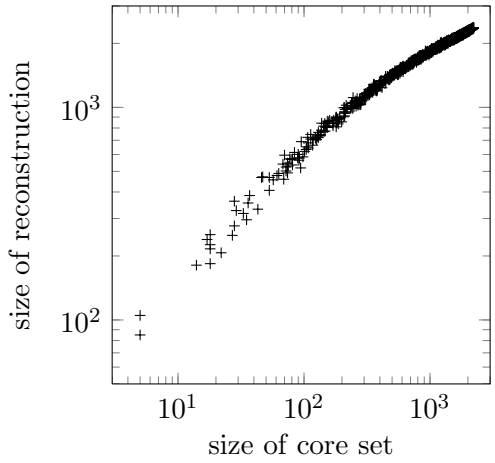

Figure 3: Applying FASTCORE on random core sets from c-Recon1

a five-fold cross-validation in which FASTCORE was used to reconstruct liver metabolism based on a reduced, randomly selected set of 80% core reactions. A hypergeometric test was used to test for the enrichment of the left-out core reactions in the added non-core reactions. Repeating this cross-validation procedure 500 times, we obtained a median hypergeometric p-value of 0.0021, indicating the ability of FASTCORE to capture missing liver specific reactions.

We also tested FASTCORE on 1000 randomly generated core sets drawn uniformly from c-Recon1, of random size drawn uniformly from $\{1, 0.9|\mathcal{N}|\}$ (where $|\mathcal{N}| = 2469$). Almost every reconstruction was obtained between 0.5 and 2 seconds (plot omitted). In Figure 3 we show the size of the reconstructed model as a function of the size of the core set, where we observe a remarkably small conditional variance. We will further investigate the latter in future work.

3.3 Reconstruction of a murine macrophage model

We also used the FASTCORE algorithm to build a cell-type specific murine macrophage model from the consistent part of Recon1bio comprising $|\mathcal{N}| = 2474$ reactions. Recon1bio ($|\mathcal{N}| = 3745$) is a modified Recon1 model that contains three extra reactions (biomass, NADPOX, and a sink reaction to balance the glycogenin self-glucosylation reaction) [5]. We used a core set comprising 300 (out of 382) proteomics derived Raw264.7 macrophage reactions, as described by Bordbar et al. [5]. (The remaining 82 reactions could not be added to the core set as they are situated in an inconsistent region of Recon 1 and therefore carry a permanent zero net flux.) For their macrophage reconstruction, Bordbar et al. used, among other methods, GIMMEp—a variant of the GIMME algorithm [3] that is similar to the MBA algorithm—and they obtained a network model containing 1026 intracellular reactions. Our main interest was to investigate whether FASTCORE can obtain a functional network that is at least as compact as the one obtained with GIMMEp. FASTCORE generated (in about one second and using 11 LPs) a consistent network model of 953 reactions, 831 of which are intracellular reactions. This is a much more compact model than the one obtained with GIMMEp.

4 Discussion

FASTCORE is a generic algorithm for context-specific metabolic network reconstruction from genome-wide metabolic models, and it was mainly motivated by requirements of fast computation and compactness of the output model.

The key advantage of having a fast reconstruction algorithm is that it permits the execution of multiple runs in order to optimize for extra parameters or test different core sets extracted from the input data [14]. For example, when working with gene expression data, the definition of the core set may depend on the threshold used to segregate between high expression genes (core reactions) and low expression genes (non-core reactions) [3]. As the choice of threshold is rather arbitrary, a practical approach could involve evaluating the robustness of the output model as a function of the chosen threshold. FASTCORE can perform this analysis in a few minutes, whereas for the same problem other algorithms would need hours or days. (Algorithms like GIMME or GIMMEp that require manual curation and assembly of subnetworks, would also fail in this kind of task.) Another example where fast computation is imperative is cross-validation. In the current study (see experiments) we ran a cross-validation procedure 500 times, an operation that took a few minutes with FASTCORE but that would barely be manageable with other reconstruction algorithms. Other examples where fast computation is important are time-course experiments or experiments involving different patients or conditions [20]. There FASTCORE could more easily identify differential models over time and/or input conditions.

Compactness is a key concept in various research areas of biology, such as the minimal genome [26, 25]. Notwithstanding, the requirement of model compactness seems to be in disagreement with the observation that biological systems are fairly redundant and this redundancy serves a specific purpose, namely, the fast adaptation to changes in the environment. Alternative pathways that perform similar functions are known to be expressed in different environmental conditions, allowing for instance to metabolize another type of sugar when glucose is not available [35]. At any rate, the pursuit of compactness in metabolic network reconstruction need not be in conflict with the notion of redundancy. Alternative pathways will be included in a reconstructed model as long as 'redundant' reactions that are supported by biological evidence are included in the core set.

Acknowledgments

We would like to thank Ronan Fleming, Nils Christian, Evangelos Symeonidis, Ines Thiele, and Rudi Balling for their feedback.

References

[1] V. Acuña, F. Chierichetti, V. Lacroix, A. Marchetti-Spaccamela, M.F. Sagot, and L. Stougie. Modes and cuts in metabolic networks: Complexity and algorithms. *Biosystems*, 95(1):51–60, 2009.

[2] R. Agren, S. Bordel, A. Mardinoglu, N. Pornputtapong, I. Nookaew, and J. Nielsen. Reconstruction of genome-scale active metabolic networks for 69 human cell types and 16 cancer types using INIT. *PLoS Computational Biology*, 8(5):e1002518, 2012.

[3] S.A. Becker and B.Ø. Palsson. Context-specific metabolic networks are consistent with experiments. *PLoS computational biology*, 4(5):e1000082, 2008.

[4] A.S. Blazier and J.A. Papin. Integration of expression data in genome-scale metabolic network reconstructions. *Frontiers in Physiology*, 3, 2012.

[5] A. Bordbar, M.L. Mo, E.S. Nakayasu, A.C. Schrimpe-Rutledge, Y.M. Kim, T.O. Metz, M.B. Jones, B.C. Frank, R.D. Smith, S.N. Peterson, et al. Model-driven multi-omic data analysis elucidates metabolic immunomodulators of macrophage activation. *Molecular Systems Biology*, 8(1), 2012.

[6] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[7] S. Chandrasekaran and N.D. Price. Probabilistic integrative modeling of genome-scale metabolic and regulatory networks in Escherichia coli and Mycobacterium tuberculosis. *Proceedings of the National Academy of Sciences*, 107(41):17845–17850, 2010.

[8] R.L. Chang, L. Xie, L. Xie, P.E. Bourne, and B.Ø. Palsson. Drug off-target effects predicted using structural analysis in the context of a metabolic network model. *PLoS Computational Biology*, 6(9):e1000938, 2010.

[9] N. Christian, P. May, S. Kempa, T. Handorf, and O. Ebenhöh. An integrative approach towards completing genome-scale metabolic networks. *Mol. BioSyst.*, 5(12):1889–1903, 2009.

[10] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.

[11] R.J. DeBerardinis and C.B. Thompson. Cellular metabolism and disease: what do metabolic outliers teach us? *Cell*, 148(6):1132–1144, 2012.

[12] N.C. Duarte, S.A. Becker, N. Jamshidi, I. Thiele, M.L. Mo, T.D. Vo, R. Srivas, and B.Ø. Palsson. Global reconstruction of the human metabolic network based on genomic and bibliomic data. *Proceedings of the National Academy of Sciences*, 104(6):1777–1782, 2007.

[13] R.M.T. Fleming, C. Maes, Y. Ye, M.A. Saunders, and B.Ø. Palsson. A variational principle for computing nonequilibrium fluxes and potentials in genome-scale biochemical networks. *Journal of Theoretical Biology*, 292:71–77, 2012.

[14] O. Folger, L. Jerby, C. Frezza, E. Gottlieb, E. Ruppin, and T. Shlomi. Predicting selective drug targets in cancer through metabolic networks. *Molecular systems biology*, 7(501), 2011.

[15] J. Gagneur and S. Klamt. Computation of elementary modes: a unifying framework and the new binary approach. *BMC bioinformatics*, 5(1):175, 2004.

[16] S. Gudmundsson and I. Thiele. Computationally efficient flux variability analysis. *BMC bioinformatics*, 11(1):489, 2010.

[17] T. Hao, H.W. Ma, X.M. Zhao, and I. Goryanin. Compartmentalization of the Edinburgh human metabolic network. *BMC bioinformatics*, 11(1):393, 2010.

[18] K. Hiller and C.M. Metallo. Profiling metabolic networks to study cancer metabolism. *Current Opinion in Biotechnology*, 24:60–68, 2013.

[19] P.A. Jensen and J.A. Papin. Functional integration of a metabolic network model and expression data without arbitrary thresholding. *Bioinformatics*, 27(4):541–547, 2011.

[20] L. Jerby and E. Ruppin. Predicting drug targets and biomarkers of cancer via genome-scale metabolic modeling. *Clinical Cancer Research*, 18(20):5572–5584, 2012.

[21] L. Jerby, T. Shlomi, and E. Ruppin. Computational reconstruction of tissue-specific metabolic models: Application to human liver metabolism. *Molecular Systems Biology*, 6(401), 2010.

[22] A.A. Julius, M. Imielinski, and G.J. Pappas. Metabolic networks analysis using convex optimization. *47th IEEE Conference on Decision and Control*, pages 762–767, 2008.

[23] N.E. Lewis, G. Schramm, A. Bordbar, J. Schellenberger, M.P. Andersen, J.K. Cheng, N. Patel, A. Yee, R.A. Lewis, R. Eils, et al. Large-scale in silico modeling of metabolic interactions between cell types in the human brain. *Nature biotechnology*, 28(12):1279–1285, 2010.

[24] R. Mahadevan and CH Schilling. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic engineering*, 5(4):264, 2003.

[25] J. Maniloff. The minimal cell genome: "on being the right size". *Proceedings of the National Academy of Sciences*, 93(19):10004, 1996.

[26] H.J. Morowitz. The completeness of molecular biology. *Israel journal of medical sciences*, 20(9):750, 1984.

[27] J.D. Orth, T.M. Conrad, J. Na, J.A. Lerman, H. Nam, A.M. Feist, and B.Ø. Palsson. A comprehensive genome-scale reconstruction of Escherichia coli metabolism. *Molecular systems biology*, 7(1), 2011.

[28] M. Pourfar, M. Niethammer, and D. Eidelberg. Metabolic networks in Parkinson's disease. In G. Grimaldi and M. Manto, editors, *Mechanisms and Emerging Therapies in Tremor Disorders*, Contemporary Clinical Neuroscience, pages 403–415. Springer New York, 2013.

[29] N.D. Price, J.L. Reed, and B.Ø. Palsson. Genome-scale models of microbial cells: evaluating the consequences of constraints. *Nature Reviews Microbiology*, 2(11):886–897, 2004.

[30] J. Schellenberger, R. Que, R.M.T. Fleming, I. Thiele, J.D. Orth, A.M. Feist, D.C. Zielinski, A. Bordbar, N.E. Lewis, S. Rahmanian, J. Kang, D.R. Hyduke, and B.Ø Palsson. Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. *Nat Protoc*, 6(9):1290–1307, Sep 2011.

[31] S. Schuster and C. Hilgetag. On elementary flux modes in biochemical reaction systems at steady state. *Journal of Biological Systems*, 2(2):165–182, 1994.

[32] T. Shlomi, M.N. Cabili, M.J. Herrgård, B.Ø. Palsson, and E. Ruppin. Network-based prediction of human tissue-specific metabolism. *Nature biotechnology*, 26(9):1003–1010, 2008.

[33] M. Sigurdsson, N. Jamshidi, E. Steingrimsson, I. Thiele, and B.Ø. Palsson. A detailed genome-wide reconstruction of mouse metabolism based on human Recon 1. *BMC systems biology*, 4(1):140, 2010.

[34] G. Stephanopoulos, A.A. Aristidou, and J. Nielsen. *Metabolic engineering: principles and methodologies*. Academic Press, 1998.

[35] J. Suckow, P. Markiewicz, L.G. Kleina, J. Miller, B. Kisters-Woike, B. Müller-Hill, et al. Genetic studies of the Lac repressor. XV: 4000 single amino acid substitutions and analysis of the resulting phenotypes on the basis of the protein structure. *Journal of molecular biology*, 261(4):509, 1996.

[36] I. Thiele and B.Ø. Palsson. A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nature protocols*, 5(1):93–121, 2010.

[37] I. Thiele, N. Swainston, R.M.T. Fleming, A. Hoppe, S. Sahoo, M.K. Aurich, H. Haraldsdottir, M.L. Mo, O. Rolfsson, M.D. Stobbe, et al. A community-driven global reconstruction of human metabolism. *Nature biotechnology*, 2013. doi:10.1038/nbt.2488.

[38] E. Vitkin and T. Shlomi. MIRAGE: a functional genomics-based approach for metabolic network model reconstruction and its application to cyanobacteria networks. *Genome biology*, 13(11):R111, 2012.

[39] Y. Wang, J.A. Eddy, and N.D. Price. Reconstruction of genome-scale metabolic models for 126 human tissues using mCADRE. *BMC Systems Biology*, 6(1):153, 2012.